



Approaches to Validating Autonomous Vehicle Safety

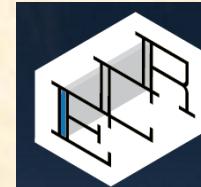
AutoSens Detroit / May 2018

Prof. Philip Koopman

Carnegie
Mellon
University



Electrical & Computer
ENGINEERING



Edge
Case
Research

Overview

■ Perception validation is the toughest part

- Brute force road testing won't get us there
- Road miles for requirements, not validation

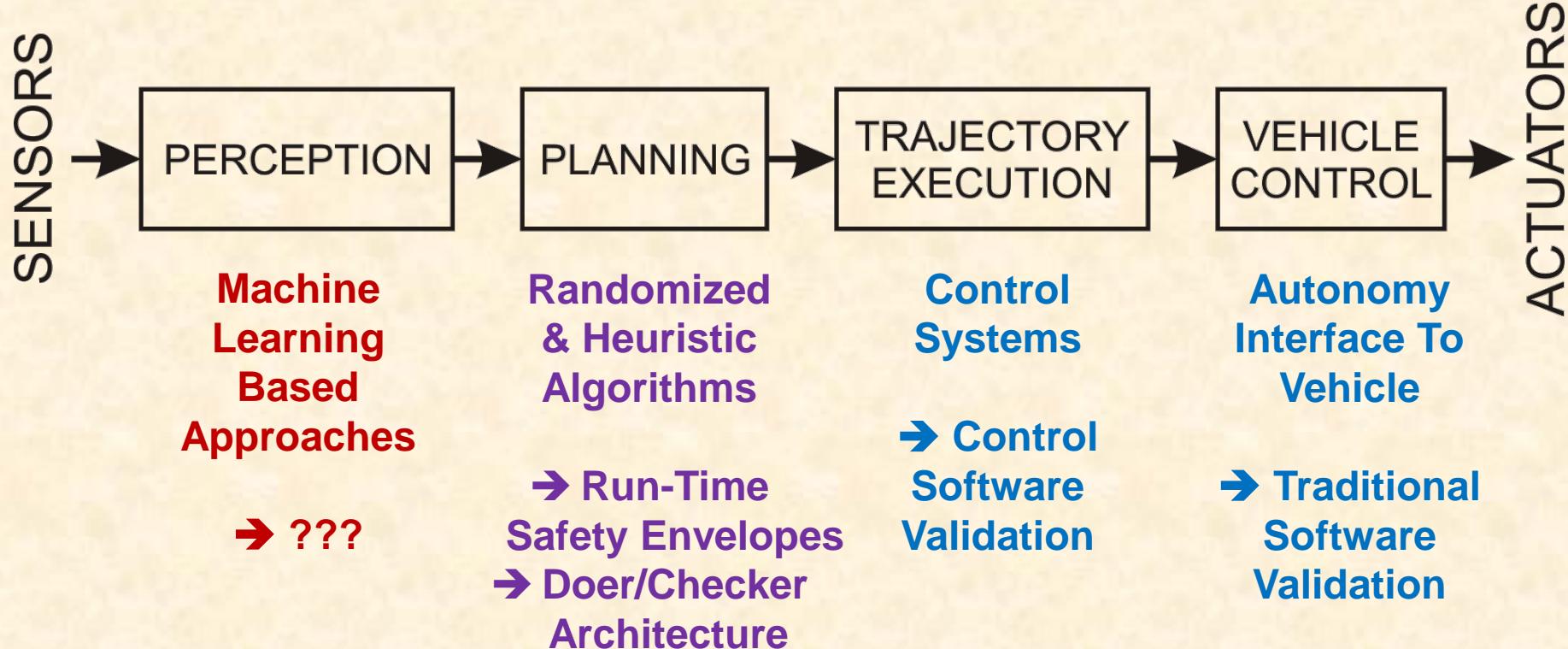


Carnegie Mellon NREC
Autonomous Platform
Demonstrator (APD)
2008-2010

■ Machine Learning is brittle

- Low cost, synthetic fault injection can improve robustness

Validating an Autonomous Vehicle Pipeline



Perception presents a uniquely difficult assurance challenge

The Tough Cases Are Legion



<http://piximus.net/fun/funny-and-odd-things-spotted-on-the-road>

<http://edtech2.boiseestate.edu/robertsona/506/images/buffalo.jpg>

<https://www.flickr.com/photos/hawaii-mcgraths/4458907270/in/photolist-Y59LC-7TNzAv-5WSEds-7N24My>

<https://pixabay.com/en/bunde-germany-landscape-sheep-92931/>

Do We Need Billions of Test Miles?

■ If 100M miles/critical mishap...

- Test 3x–10x longer than mishap rate
→ Need 1 Billion miles of testing

■ That's ~25 round trips on every road in the world

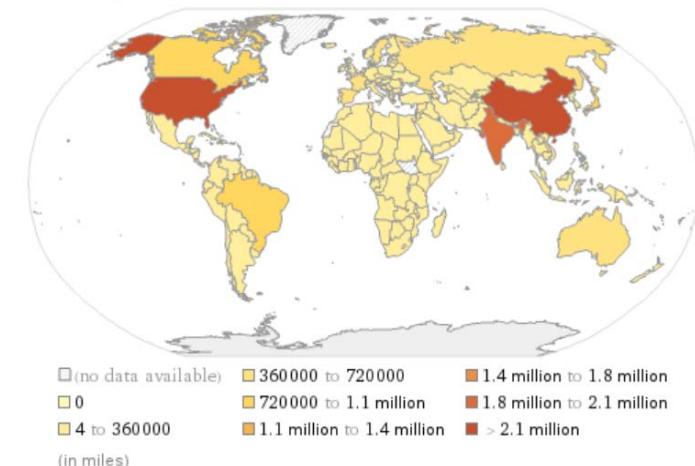
- With fewer than 10 critical mishaps

...

- *Then you're only as good as a human*
– (Including the impaired humans!)



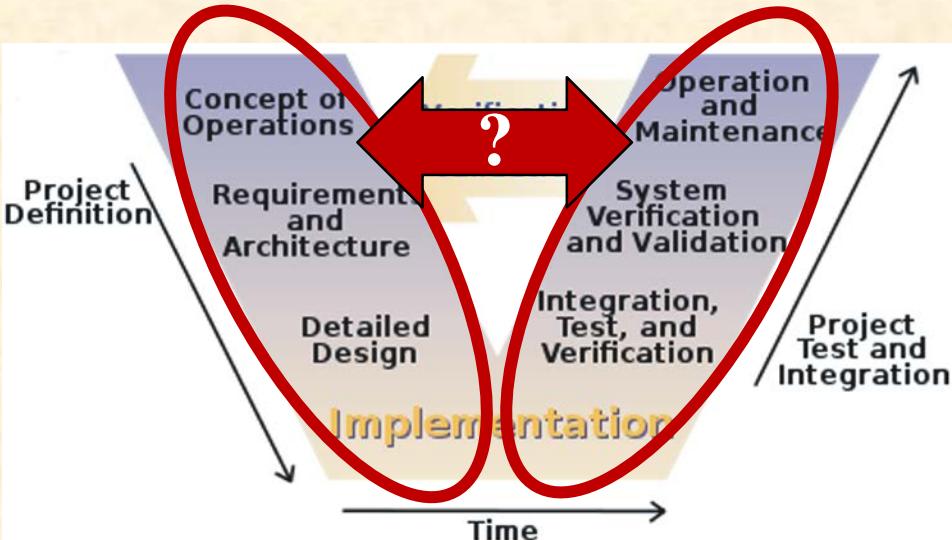
Total road length map:



Traditional Validation Meets Machine Learning

■ Machine Learning (inductive training)

- No requirements
 - Training data is difficult to validate
- No design insight
 - Generally inscrutable
 - Prone to over-fitting/gaming



Road “Testing” Should Gather Requirements

■ Use road miles to gather requirements

- Novel objects, events, scenarios (OEDR-centric)
- Novel operating conditions (ODD-centric)
- Edge cases that present problems



<https://goo.gl/3dzguf>

■ Think “requirements testing” not “vehicle testing”

- Disengagements are a blunt instrument for detecting novelty
- Look for novelty even if vehicle “test” is passing



[https://en.wikipedia.org/wiki/Magic_Roundabout_\(Swindon\)](https://en.wikipedia.org/wiki/Magic_Roundabout_(Swindon))



<https://goo.gl/J3SSyu>

Smart Use of Simulation

■ Point of view: everything is a simulation

- Software & HW component simulation
 - Includes sensor performance characterization
- Software vehicle simulation
- HIL testbeds
- Closed course testing
 - Simulated environment, obstacles, events
- Public road testing
 - Assumes representativeness



University of Michigan

■ Even a “perfect” simulation needs scenarios as inputs

- You need a test plan that covers all required functionality

Simulation As Risk Reduction

**All Simulations Are Wrong...
But some simulations are useful**

■ It's all about the assumptions

- “Perfect” simulation is expensive
- Exploit the cost/fidelity tradeoff

■ Layered Strategy:

- Simplifications for large search spaces
- Complex simulations for residual risks
 - Validate assumptions
 - Look for emergent effects and surprises

■ Use road tests to validate simulations

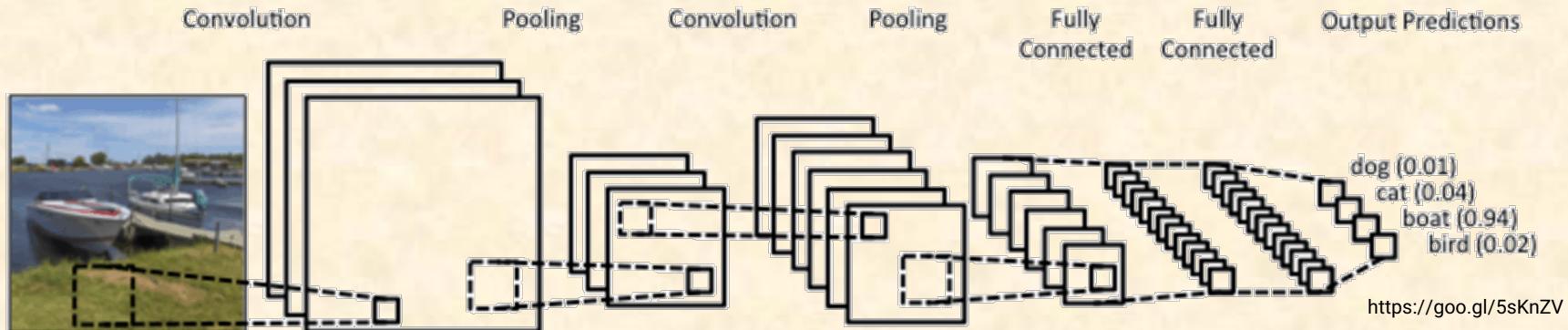
- Identify and concentrate simulation residual risks...
 - ... which you get from analysis and road experience

High Fidelity Simulation/Test; Look For Requirement Gaps

Validation Phase	Residual Risks Resolved	Residual Threats to Validity
Deployed Vehicle	- Safety argument assumption violations detected	Unexpected Scenarios and ODD violation modes
Pre-Deployment Road Tests	- Scenario Exceptions - Environment Exceptions	Unencountered, novel scenarios/environment
Closed Course Testing	- Un-modeled sensor degradation - Environment sim. artifacts - Environment sim. assumptions	Requirements gaps in human behavior, road hazards, etc.
Vehicle Testing with Simulated Environment	- Vehicle simulation artifacts - Vehicle simulation assumptions	Environmental simulation quality
High-Resolution Vehicle & Environment Simulation	- Simulation artifacts/assumptions - Emergent behavioral interactions	Vehicle simulation quality
Simplified Vehicle and Environment Simulation	- Vehicle dynamics - Sensor data quality - Actuator effects	Simulation simplifications and assumptions
Subsystem Simulation	- Emergent component interaction	Multi-subsystem interactions
Software Validation	- Ordinary software defects	Emergent behaviors & ML

Low Fidelity Simulation; Look for Design Defects

ML Is Brittle To Adversarial Attacks



QuocNet:

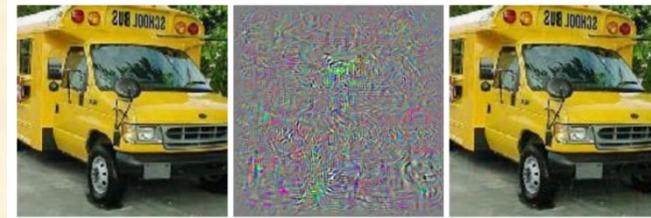


Car

Not a
Car

*Magnified
Difference*

AlexNet:



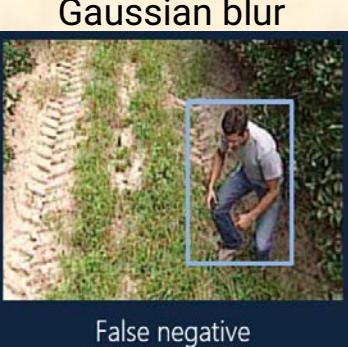
Bus

*Magnified
Difference* Not a
 Bus

Also Brittle To Physics Based “Attacks”

■ Sensor data corruption Experiments at NREC

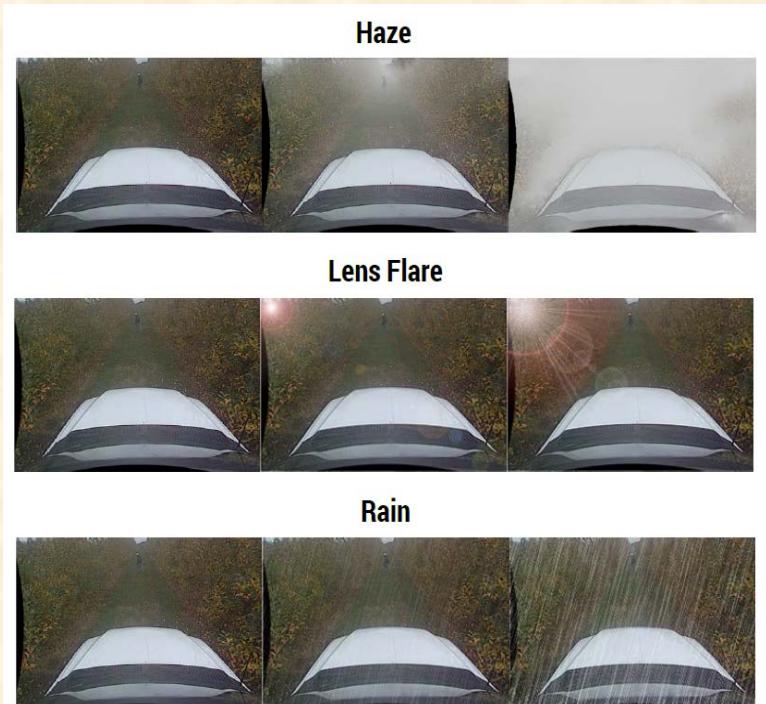
Synthetic Equipment Faults



Gaussian blur

Exploring the response of a DNN to environmental perturbations from “Robustness Testing for Perception Systems,” RIOT Project, NREC, DIST-A.

Synthetic Environment Robustness Testing



Haze

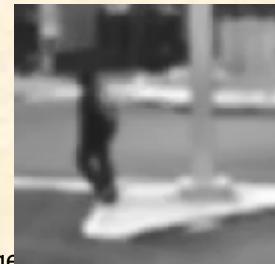
Lens Flare

Rain

Sensor Fault Injection

■ Synthesize faults during development

- Degrade sensor inputs
 - Use simple approximate models
- Insert slightly malformed objects
- Even “unrealistic” faults provide insight

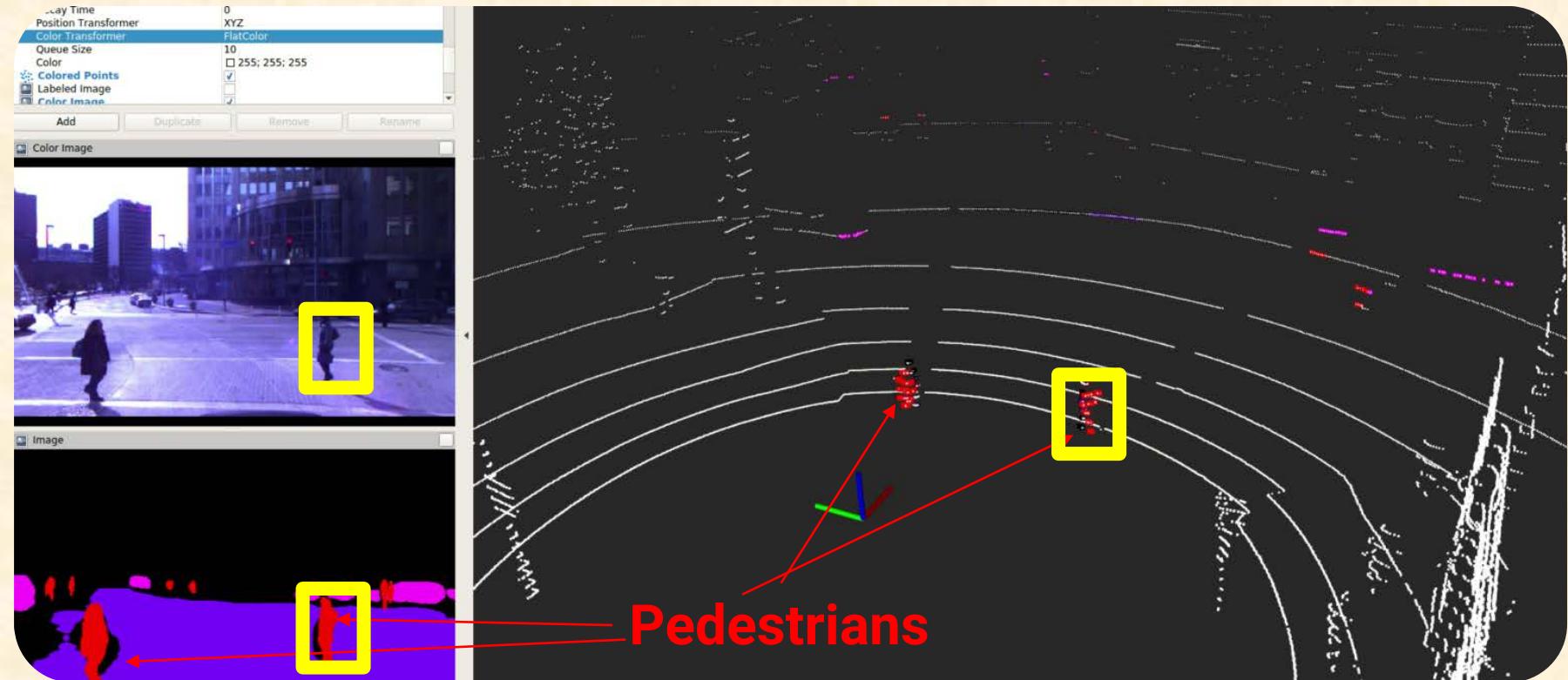


■ Realistic data for validation

- Road data when you have it
- Highly realistic synthetic faults
 - E.g., photo-realistic haze

Multiple Sensors

- Fault injection needs to coordinate multiple sensor inputs



Ideas for Better AV Sensor Validation

1. Concentrate on data collection with road miles

- Look for things beyond disengagement triggers
- Road tests should validate, not debug

2. Use a layered approach to simulation

- Exploit fidelity/cost tradeoffs
- Validate assumptions & simplifications

3. Sensor-level fault injection can help

- Generative models to create test scenarios
- Fault injection to improve robustness



Real image



Synthetic image